

SYSTEM AND METHOD FOR VLSI VISUALIZATION

Field of the Invention

5       The invention relates to visualization as an aid to the design, verification, and test of integrated circuits.

Background of the Invention

10      As integrated circuits (IC) become increasingly dense and complex, all aspects of design, including custom and synthesized design, design verification and testing are becoming more and more difficult. The immensity of the data and the complex interrelationships among the various  
15      aspects of design including the design data, technology, and test, make it extremely difficult for IC designers to complete a design with confidence that it will function at the intended performance, and be manufacturable and reliable. Design aids that enable the visualization of  
20      circuit functionality, simulated under various operating conditions, process conditions, and electrical input conditions, that are interactive and cross-probeable with the device and circuit elements of the design views would greatly enhance the ability to create robust designs  
25      more effectively and in less time. This invention addresses this need by providing a system and method of visualizing the electrical activity and/or logical activity of an IC. It is easy to use and interpret, and the various design, simulation, and hardware data views contain parameterized, cross-probeable and interrelated content, at the transistor, gate, or circuit level.  
30      Here, the term "visualization" is used broadly, and

5 include animation (for example, slow motion movies) with or without audio enhancement, plots of interrelationships between various dependent and independent variables, and also includes tactile outputs to assist the visually impaired.

A particular embodiment that relates a simulated view to a photon emission view and provides comparative visualizations of each, is disclosed herein. This  
10 particular embodiment relates to the commonly owned patent application no. \_\_\_\_\_ (IBM Docket YO998-343) entitled METHOD FOR VLSI SYSTEM DEBUG AND TIMING ANALYSIS filed herewith, the disclosure of which is incorporated herein by reference. This patent  
15 application no. \_\_\_\_\_ (IBM Docket YO998-343) provides a means by which to characterize internal IC switching activity by measuring and presenting photon emission data according to the technology described in the commonly owned and co-pending U.S. patent application serial no. 08/683,837, entitled NONINVASIVE OPTICAL  
20 METHOD FOR MEASURING INTERNAL SWITCHING AND OTHER DYNAMIC PARAMETERS OF CMOS CIRCUITS, filed July 18, 1998, the disclosure of which is also incorporated herein by reference.  
25

A feature of simulated picosecond imaging circuit analysis (PICA) is that any test or instruction set sequence may be applied to the circuit under investigation and will lead to observable results. This  
30 is contrasted to the present state of PICA measurements. Care must be taken to keep the PICA measurement loop short, and the test loop must be exercised repeatedly.

This is because of the low level of emitted light in the photoluminescence process. This low light level, and the finite duration of the test loop make PICA measurement acquisition times slow. Thinning of the chip further increases the complexity of preparation of test. The simulation, however, is not limited by the low probability for photoluminescence, and any arbitrary test sequence or instruction sequence may be applied and the reaction visualized. The requirement to repeatedly exercise the circuit disappears for the case of PICA simulation. Simulation is also quick compared to measurement. Simulation is also amenable to characterization of the design prior to building the device. Hence, PICA simulation opens up many opportunities for improving designer productivity which have been heretofore unavailable by other means. An example of such a newly enabled application is a test coverage checker which produces output in a visual, intuitive manner.

The invention also relates to the problem of expressing causal relationships in a manner which is easily and intuitively interpreted. For example, circuit designers often analyze simulated circuit activity by plotting voltage waveforms on a common time base to visualize the causal relationships, such as shown in Figure 1 for an inverter chain. The dotted lines pointing from one waveform to the next indicate the causal order of switching events. These relationships can be expressed in terms of a "sequence graph". Figure 2 shows an example of a sequence graph, 12, for the invertor chain. The net names (1, 2, 3, etc.) are indicated at the nodes,

as are the corresponding waveforms (a, b, c, etc.). The arrows connecting the nodes indicate the causal relationship between the voltage behaviors at the nets.

5      Summary of the Invention

It is an objective of this invention to provide a system and method to visualize the simulated internal functioning of ICs for effective design evaluation, 10 design of test, and comparison to the measured internal functioning of ICs as measured using techniques such as the newly available photon emission measurement also known as PICA. An output of the PICA system is a visual representation of transistor switching events, in the form of a slow motion video. The present invention is related to visualization of simulated IC activity. One 15 form of activity further described in the preferred embodiment is the simulated version of the PICA slow motion "movie". This has multiple uses, including i) predicting and thereby preparing test and analysis approaches for PICA measurement, ii) learning through visual study of the switching behavior aligned to the physical layout or optical microscope image, iii) flagging of improper behavior such as incorrect logical 20 operation occurring during certain ranges of simulation conditions, and iv) comparison of the visualizations 25 between simulation and measured data sets.

The structure comprising the system and the method of use 30 apply not only to the visualization of simulated emission images but apply to other simulated design data views as

well. Examples are switch level simulation, current density simulation, and power density simulation.

The preferred system and method described herein use a sequence graph which represents the causal activity induced upon a circuit network by a stimulus. This sequence graph may then be annotated with the visualization representation, such as PICA emissions at devices connected to the elements of the graph.

Likewise, other visualizations such as current, local power dissipation, local noise, etc., may be annotated in this way.

#### Brief Description of the Drawings

Figure 1 illustrates voltage waveforms corresponding to a delay chain of inverters, with the causal relationships indicated.

Figure 2 illustrates a sequence graph for an inverter chain (top), and corresponding schematic (bottom).

Figure 3 illustrates a switching visualization system, structures comprising the system, and the flowchart which shows how the elements of the system are used together in the method.

Figure 4 illustrates the annotation structure of an annotated sequence graph file.

Figure 5 illustrates the flow of a photon emission simulation.

Figure 6 shows areal images of the simulated emission corresponding to a physical design during two time intervals.

5       Figure 7 illustrates a tool that compares simulated data to measured emission data.

Figure 8 illustrates the operation of an optical emission simulator.

10      Figure 9 illustrates a method for comparing simulated emissions to measured optical emissions.

Detailed Description of the Preferred Embodiments

15      The preferred embodiment provides a system, structures, and method by which visualization of the simulated circuit behaviors and properties is performed within an IC CAD viewer. These visualizations are viewable within  
20      the IC viewer along with the regular device views and measured IC data, verification traces, logical data, and test data. In the case of electrical simulation, currents simulated at nets can be computed to form PICA representations, time varying power maps at the  
25      individual transistor and net level, noise, and electromigration maps, to name a few types of visualizations. The novelty of electrical simulation visualization is the viewing of complex circuit activity in an intuitive fashion, down to the individual device  
30      level. This can be related to measured circuit activity at the device level by methods such as PICA. As mentioned in the background, there are situations where

simulated PICA is more accessible than the physically measured PICA for the analysis of design because there are no constraints on the test for simulation, and simulation is fast compared to measurement. When

5 visualizing the physical responses of the elements comprising the device, such as nets and transistors, the visualization computation is very efficient because of the compact nature of the response data in the form of a "sequence graph" which contains the representation of the

10 responses including connectivity and causality. The visualization engine need not search the entire dataset of responses from elements of the device, but only those that are influenced by the stimulus. The relationships between the elements, including the cause and effect

15 relationships, and pointers to the desired data to enter visualization, such as voltage waveforms or PICA emissions, are all included in this "sequence graph" representation.

20 It is noted that visualization may take place at any arbitrary design hierarchy, including mixed hierarchy. For example, for PICA visualization, it is natural to view the emissions on top of the flattened physical design view. For viewing logical activity, one might

25 visualize activity of logical blocks which may be comprised of more than one leaf cell. The desired hierarchy, or mixture of hierarchy, may be defined by interaction with the user interface.

30 figure 3 illustrates the switching visualization system, structures comprising the system, and the flowchart which shows how the elements of the system are used together in

the method. Generalized input, 100, is expressed in terms of a *sequence graph*, 30. For the preferred embodiment, this input is the output of a circuit simulator, such as a SPICE simulation. This is shown 5 inside the dotted box, 101. Stimuli such as testvectors, 10, are applied to the circuit netlists, 20. The circuit simulator, 15, provides outputs in the form of waveforms or currents at the desired netlist elements.

10 Device activity traces are another type of generalized input, 100, shown in the Figure. This is illustrated in the lower dotted box, 102. These traces may be measured, which is the case of the example shown in the Figure, or they may be simulated. For measured traces, the inputs 15 are instruction sequences which are applied to the device (state machine). The measured outputs are "traces" which are a sequence of logical states (0's and 1's) at designated register bits or other designated elements. These traces are then expressed into the sequence graph 20 format.

For the case of simulation, stimuli such as instruction set sequences or testvectors are applied to the network under analysis, to produce traces at designated trace 25 locations such as register array bits. The simulation may be a logical simulator or a circuit simulator. There are advantages to using both simulation methods and visualizing each view. In each case, the trace application input would consist of a sequence of 0's and 30 1's indicating the register bit states at each cycle of the state machine under analysis. The visualization provided by the system may consist of a physical mapping

of the logical states on the physical design data, schematic design data, or logical data. Additional internal "probe" points of the network may be designated for simulation output for the sake of visualization.

5 Then, the activity leading to the traces may be visualized in logical or waveform formats. A circuit simulator would provide *voltage* or *current waveforms*, in which the trace application would proceed as previously described.

10 For *debugging*, the progression of logical states, and wire delays visualized together may help path analysis and optimization. A further application is the checking of the instruction traces themselves. One could check

15 the logical progression of the instruction trace through the device to visualize if the instruction trace is indeed exercising the device as intended. For example, a legacy instruction trace may not be effective in exercising a new machine implementation and in this case,

20 the differences may be visualized intuitively. Certain measures could also be derived for the instruction trace such as: relative utilization of elements, power dissipation, noise generation, etc. Each of the aforementioned measures may be derived at the local

25 element level, or at the global level. For example, it would be useful to obtain a value for the power dissipated when a certain instruction sequence is run.

Information from the instruction trace dependence on power may be used to characterize the device power dissipation. This information could be used for compiler optimization for power, for example. This simulation 30 would provide a far more accurate estimation for power

than the switching factor estimates largely in use today. Again, these computations and visualizations become traceable with the compact connectivity and causality expression provided by the sequence graph notion.

5       Similarly, traces may be created to maximize noise. The work to create suitable instruction traces for noise tolerance test may be done by simulation. The optimized instruction trace, or testvector sequenced derived from the instruction trace or from the sequence graph, would  
10      be utilized in the IC hardware test. In the latter derivation of a testvector sequence to mimick the instruction set, comparison of the sequence graphs for each case would provide a measure of how well the testvector recreates the activity generated by the  
15      instruction trace. A further application might maximize skew. Skew tolerance measured through simulation could then be used to tune wires to improve the skew tolerance of the device as necessary. A wire optimization tool may even be programmed to take as input, the annotated  
20      sequence graph, in this case annotated with timing data from the simulations or measurements. Similarly, testvectors may be verified and debugged or characterized in much the same manner.

25      The reaction of the circuit to the system input, 100, may be represented in the sequence graph, 30. The sequence graph is derived from the netlist or schematic, and comprises a record of the events that occurred within the network as a result of the system input. For the case of  
30      the digital circuit schematic network, the sequence graph may be a cone of logic with base at the input(s) which propagates signal to the outputs, latch points, or other

points of interest. Figure 2 shows a sequence graph (top) for the simple case of an inverter chain. The corresponding schematic is shown on the bottom of the Figure. The nodes of the sequence graph represent the  
5 states at the nets (n1 through n7 of the schematic), with the leftmost node corresponding to the input. The nodes are also annotated (a,b,c, etc.) with identifiers for the corresponding voltage waveforms of Figure 1.

10 Figure 4 depicts the structure of a sequence graph. Again, the inverter chain is used to illustrate the point. The node is annotated with the element to which it corresponds. For the first node, the element is net 1. There is also a pointer to the address for the  
15 corresponding state variable, in this case, the voltage waveform, 210, at net 1, which is labeled "a". There is also a reference to the next state(s), represented by the arrow between the nodes of the graph illustration. Other variables, such as optical emission waveforms, 220, or  
20 transition sequences, 230, can be derived from the state variables (as depicted by the derivation engine, 35, in Figure 3). The Figure shows that the graph may also contain pointers to one or more derived variables. If convenient, separate sequence graphs for each variable  
25 type, or set of variable types, may be retained.

The appropriate technology model (in this case, optical emission waveforms can be derived from the current waveforms and the optical emission model) is provided as  
30 input to the derivation engine. The output of the derivation engine are the derived variables, and these may be annotated upon the sequence graph to form an

annotated graph, 50. Going back to Figure 4, 230 depicts logical transition state data as another example of a derived variable. Other possible derived data are local noise fluctuations, power line bounce, local power  
5 dissipation, and so forth. These comprise the annotated sequence graph, 50, Figure 3. The technology model input to enable derivation of the derived variables is depicted by 40 in the Figure. The Figure depicts the sequence graph and annotated sequence graph as separate entities,  
10 generated separately, to simplify teaching of the separate activities and computations involved. However, it is obvious to one skilled in the art that the delineation of the sequence graph and annotated sequence graph may be broadly defined and still come within the  
15 scope of the invention. It would be convenient to annotate the original sequence graph with the state variables, or pointers to the state variables.  
Subsequent computation would result in the derived  
variables, which would conveniently be annotated to the same sequence graph, to comprise the annotated sequence graph.  
20 The original representation of the sequence graph may be retained, or discarded at any point in the process.  
25 In another related implementation, still consistent with the spirit of the embodiment shown in the Figure, multiple sequence graph files are generated and retained, should multiplicity be convenient in the visualization flow. In the latter case, the appropriate pointers between the variables relevant to the various  
30 visualizations would facilitate the concurrent visualization of two or more circuit response

representations. A graphical user interface and data structure that supports toggling between two or more views would be valuable to the designer to intuitively relate two representations of the circuit responses. For  
5 example, viewing of the circuit activity (e.g., voltage, current, or optical emission, etc.) in the physical (layout or physical image) view would be enabled concurrently with the schematic view visualization. Here, the graphical user interface would provide  
10 functions or **stop, forward, backward, step, integral**. "Step" provides function to step through the "frames" of the animation, and "integral" displays the entire set of activities demonstrated. The latter could be the computed integrated emission in the case of optical  
15 emission, or could be a total switching activity visual report in other views (for example, the schematic view). The latter would provide an intuitive rendering of the cone of logic influenced by the stimulus (or subset, thereof - not all influenced devices would necessarily switch). Furthermore, the causal relationships may be  
20 illustrated in the static or dynamic visualizations. For the integrated view, it would be especially useful to have pointers such as arrows to depict the causal relationships. While causality might be inferred in the  
25 case of a schematic view, arrows depicting causality would clarify the dependencies when multiple signal paths may influence a particular gate. These pointers, or arrows, would be turned on or off by the user. An objective of this inventions is to supply the designer  
30 with views of the circuit response with the aid of *intuitive visualizations which may be combined as desired and be presented in as simple or in as much detail as*

desired, so that he/she may best interpret the circuit activity.

Returning to Figure 3, the annotated sequence graph, 50, and visualization models, 60, and the layout vs.

schematic file (to recognize the devices/nets which are relevant to the visualization) are input to the visualization engine, 55. The visualization models comprise those relationships that are needed to compose the visualization, 70. An example of a visualization model for optical emission visualization are the emission models, which compute optical emission as a function of space and time in the vicinity of an emitting transistor. These models, together with the annotated sequence graph content, and visual backdrop, 65, comprise the inputs to the visual simulation engine, 70. Having on hand the relevant time varying data (those state variables and derived variables) and causal relationships between them, mappings to the circuit elements, and visualization models to render these variables against a backdrop, a richness of visualizations become possible. The resulting renderings work dynamically and interactively with the graphical user interface to vary the time and spatial (such as region, rotation, etc.) domains, and traverse design hierarchy as desired.

The rendering of the various views is finally visualized at an output device, 80. Here, a designer's workstation is depicted with the renderings within an IC viewer shown on the display. It is obvious that the output device as shown may be replaced by a variety of clients and still be consistent with the invention. depicted on the

display (front) is a rendering of an optical emission in the layout view, and in the schematic view (back).

Figure 5 again uses the simple example of an inverter chain to illustrate how circuit activity may be visualized in the schematic view, in the form of an animation or a still, integrated, image. Here, the individual transistors are explicitly drawn in the circuit schematic, 500. In 510, transistor nFET2 is shown "lit up" to represent that the transistor is switching, and/or is conducting current, and/or emitting light. In 520 pFET2 is shown "lit up". This would occur a number of "time frames" after nFET2 is active, but the Figure only depicts the frame that the emission is at its maximum. 530 depicts as integrated view, with all of the transistors "lit up", and arrows directed between them to indicate the causal order of events.

Figure 6 illustrates, again using the simple example of an inverter chain, optical emission activity overlayed on a layout view. Here, for simplicity, the MOSFET device regions are indicated by rectangles. Those familiar with the present state of PICA activity will know that PMOSFETs emit much less than nMOSFETs. For simplicity, pMOSFET and nMOSFET emission strength is not differentiated in the illustration. 610 illustrates nFET2 "lit up" in this layout view, 620 illustrates the next event, where pFET2 "lights up", and 630 illustrates the integrated emission where all of the active transistors are "lit up", and arrows between them are shown to indicate the causal ordering of events.

Another rendering is a current flow visualization, as shown in Figure 7. Element 710 illustrates, in the schematic view, current flow along the net connecting two inverters. Element 720 shows current flow depicted in a physical design view using the imagery of a lump traveling through the network. Here, the area of the lump could be colored to highlight the charge motion and to differentiate it in the still view, with a physical structure. The current could also be depicted as a bright dot moving along the network for each net carrying a current, for example.

Another rendering is a voltage visualization. Here, wires and transistors would be the backdrop, and the activity would be the motion of a voltage waveform moving through the network. A net, or portion of a net, would be colored, for example, and deepen as it reaches a voltage rail. Vdd and ground could be represented by differing colors.

Another rendering is the charting of the sequence graph, or annotated sequence graphs or derivatives of, themselves. The sequence of causal relationships of a nontrivial network may be plotted such as 12 of Figure 2, to show the causal ordering of events. Several paths may be shown on one chart to illustrate parallel path behavior.

It is noted that complex circuits, and complex circuit types benefit from analysis of the subject visualization system and method. For example, *asynchronous circuits* are a particular challenge for timing analysis as well as

state analysis. The detailed timing behavior and logical state behavior is difficult to characterize since the usual clock boundaries and rules do not apply. The present invention provides means of characterizing 5 asynchronous circuits in detail, retaining all of the causal relationships that form the network reaction to a given stimulus. The state variables and derived variables may be analyzed, and the sequence graph, or annotated sequence graph may be directly analyzed for 10 consistency with expected behaviors (rules expressed in terms of the causal relationships).

Those of ordinary skill in the art will understand that with this system, numerous other visualizations of 15 circuit activity may be rendered within the scope of this invention. These may include noise simulation, electromigration fault simulation, power dissipation simulation, and so forth.

It is noted that data reduction and management are 20 important for practical implementation. Compact representation of the transistor level data uses device recognition (by layout vs. schematic assignment).

It is further noted that this data may again be processed 25 to emphasize differences, golden image comparisons, etc. It is further noted that the various visualizations may be brought up in any IC viewer, and may be cross-probed between themselves or the design, verification, or test 30 views, as appropriate.

A particular embodiment is a substructure which enables the visualization of simulated circuit functionality in slow motion time, one form of which is an animation of the optical emissions which occur as a result of a  
5 certain input set of stimuli. While this can be expressed in the general system and method of Figure 3, in this particular case, a direct "hardwired" route to the animations and still imagery is effective. The optical emission means of representing switching events  
10 in the animation is a natural presentation of the events for studying and interpreting circuit behavior, and is amenable to the "motion picture" medium. Figure 8 illustrates the process flow, starting with (110) which presents the simulation results in terms of currents  
15 and/or voltages ( $I(t)$  and/or  $V(t)$ ) describing the circuit activity, and (120) is the photon emission simulation result,  $F(I(t), V(t))$ , for each emitting transistor. The coordinates of the emitting transistors are located and the photon emission at each transistor is overlayed with  
20 the layout view. Here, the spatial variation at each transistor is computed in accordance with a physical model which assigns intensity to the emissions in the spatial regions in the vicinity of the transistor channel regions. The physical assignment might be a Gaussian  
25 spreading function applied to point sources located at preassigned "pixel regions" at and in the vicinity of the transistor channel. A Monte Carlo simulation may also be exercised to create a randomized distribution of photon events. The simulation visualization has been described  
30 in the copending commonly owned patent application entitled "METHOD FOR VLSI SYSTEM DEBUG AND TIMING ANALYSIS". The simulated emissions are overlayed over

the physical design data to aid in the visualization and interpretation of the emission data. A photomicrograph image of the region of interest might also be used for the overlay if desired. Images such as these can be combined in time sequence to make a slow motion movie of circuit activity. It is noted that this particular embodiment may be implemented with or without the aid of a sequence graph for the end implementation.

Once the simulated emissions are computed and areally interpreted as described above, comparison may be made to measure optical emissions as a diagnostics or comparative analysis technique. The method is illustrated in Figure 9. The simulated and measured emissions, (910) and (920), respectively, are compared and analyzed for differences by the comparative analysis tool (930). The output of (930) is a representation of the errors and faults (940). The concept of comparative analysis between simulated and measured results may be generalized to other visualizations such as current and voltage measurements in a similar fashion.

In each case the objects being compared need not be limited to areal images. In the case where a sequence graph exists this may be compared to the measured data in several different ways. A sequence graph corresponding to the measurement may be derived and compared to the simulation, or the simulated sequence graph may be compared to the measurement by one to one correspondence comparison between derive emissions as interpreted from each representation. It is not necessary to have complete correspondences between each emission. For

instance, an emission may be predicted in the simulation that is below observability of the measurement and is thereby not observed in the measurement. However, if an emission is observed in the measurement that is not observed in the simulation, a fault may be detected.

5